CS493.71

Project 1

Database Design

Favian Flores & Md Uddin

**Query 1:**

Rationale:

For the first query we decided to use Mongo. Since speed is a priority, document-based databases, such as Mongo, tend to provide better performance than relational databases because all the information is kept together. It also allows for more flexibility because each piece of data does not need to have the same number of fields. Once the data is loaded into the server, the query is simply pulling up the relevant document and no further processing is required. Overall Mongo seems the best choice for the data given and the query.

Implementation:

We decided to store each disease in its own document along with all relevant information. Mongo assigns a unique id to each document so the database should respond quick.

**Document:**

Id: string

Name: string

Treats: array

Palliates: array

Gene: array

Location: array

**Query 2:**

Rationale:

For the second query we decided to use neo4j. Query 2 requires finding connections between data so a graph-based database seems appropriate. Neo4j databases use relationships and have several options which aid in the finding of these relationships. While document-based or relational databases are appropriate for storing large amounts of data, they don’t perform as well as graph databases when the primary purpose it to find relationships. Since speed is also a priority for this query, neo4j seems the most appropriate choice.

Implementation:

We decided to replicate the node/edge model of hetionet and then query the edge paths. Since this database requires computation for each query, it was important to use the options neo4j offers in order to speed up the query. We applied constraints asserting that the id for each node is unique allowing the database to locate each node faster. We also used certain modifiers when matching, such as WHERE NOT, and DISTINCT which cleaned up the data we received before the information is returned to the user.

**Node:**

Name: string

Id: string

**Edge:**

Relationship: string

Source: Node

Target: Node